

Домашние задания по курсу «Математическая логика» ИТМО, группы 2537-2539, осень 2013 г.

Задания с дробными номерами необязательны, их решение, впрочем, будет давать бонус для зачета. Для всех программ кодировка выходных файлов должна быть либо CP1251, либо UTF8.

- 1 Написать программу, проверяющую доказательства в исчислении высказываний на корректность. Входной файл представляет из себя последовательность высказываний, по высказыванию на строку. Высказывания удовлетворяют приведенной ниже грамматике.

$$\begin{aligned}\langle \text{выражение} \rangle &::= \langle \text{дизъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle ' \rightarrow ' \langle \text{выражение} \rangle \\ \langle \text{дизъюнкция} \rangle &::= \langle \text{конъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle ' \mid ' \langle \text{конъюнкция} \rangle \\ \langle \text{конъюнкция} \rangle &::= \langle \text{отрицание} \rangle \mid \langle \text{конъюнкция} \rangle ' \& ' \langle \text{отрицание} \rangle \\ \langle \text{отрицание} \rangle &::= ('A' \dots 'Z') \{ '0' \dots '9' \}^* \mid '!' \langle \text{отрицание} \rangle \mid '(' \langle \text{выражение} \rangle ')'\end{aligned}$$

Результатом работы программы должен быть либо текст:

Доказательство корректно.

либо указание на ошибку — текст вида:

Доказательство некорректно начиная с высказывания номер $\langle \mathbb{N} \rangle$.

где $\langle \mathbb{N} \rangle$ — это номер строки, содержащей первое недоказанное высказывание.

- 2 Написать программу, преобразующую вывод $\Gamma, \alpha \vdash \beta$ в вывод $\Gamma \vdash \alpha \rightarrow \beta$ (либо уведомляющую, что исходный вывод некорректен). Первой строкой входного файла должна являться строка, перечисляющая гипотезы, использованные в выводе, и выводимое утверждение. На следующих строчках входного файла перечислены высказывания исходного вывода. Высказывания удовлетворяют грамматике из предыдущего задания, первая строка соответствует следующей грамматике:

$$\langle \text{заголовок} \rangle ::= \{ \langle \text{выражение} \rangle ', ' \}^* \langle \text{выражение} \rangle ' \mid - ' \langle \text{выражение} \rangle$$

Символ $' \mid '$ имеет ASCII-код 124 (десятичное).

Результатом работы программы должен быть текст, содержащий преобразованный вывод. Формат выходного файла совпадает с форматом входного файла. Вы можете предполагать, что входной файл содержит корректный вывод требуемой формулы.

- 3 Написать программу, строящую доказательство указанного во входном файле высказывания (если оно общезначимо), либо дающую оценку пропозициональных переменных, на которых высказывание ложно (если оно опровержимо).

Входной файл состоит из единственной строки, содержащей формулу исчисления высказываний, которую требуется доказать или опровергнуть. Высказывание удовлетворяет грамматике из первого задания. Выходной файл должен либо содержать доказательство высказывания (в формате доказательства из первого задания), либо содержать фразу, удовлетворяющую грамматике:

$$\begin{aligned}\langle \text{строка} \rangle &::= \text{'Высказывание ложно при ' } \langle \text{назначение} \rangle \{ ', ' \langle \text{назначение} \rangle \}^* \\ \langle \text{назначение} \rangle &::= \langle \text{переменная} \rangle ' = ' ('И' \mid 'Л')\end{aligned}$$

Например, при входной формуле **!A&!B** результат (с точностью до порядка переменных и конкретного контрпримера) должен выглядеть так:

Высказывание ложно при A=И, B=Л

- 3.5 Написать программу, проверяющую общезначимость в интуиционистской логике высказывания, указанного во входном файле, и строящую опровергающую его модель Крипке, если оно не общезначимо.

Входной файл — единственная строка, содержащая формулу исчисления высказываний. Выходной файл — либо единственная строка

Высказывание общезначимо в интуиционистской логике.

либо запись опровергающей его модели Крипке в следующем формате.

Файл состоит из некоторого количества (n) строк, i -я строчка файла содержит описание мира W_i . Миры должны быть топологически отсортированы: если $W_i > W_j$, то $i > j$. При этом все миры, большие данного, должны идти непосредственно за ним, одним куском: для любого i найдется такой k , что $W_i < W_j$ тогда и только тогда, когда $i < j \leq k$. Описание представляет из себя символ звёздочки, за которым идет список имен переменных, которые вынуждены в данном мире, указанных через запятую. Все миры W_j , такие что $W_j > W_i$, должны быть описаны в последующих строках, с отступом от позиции мира W_i . Все миры с одинаковым отступом считаются братьями. Например:

```
*
* P
  * P,Q
  * P,R
* Q
```

В данном примере $W_1 < W_2, W_3, W_4, W_5$, также $W_2 < W_3, W_4$. Также $W_2 \models P$, $W_3 \models Q$, $W_4 \models R$, $W_5 \models Q$. Да, этот пример опровергает высказывание $P \vee \neg P \vee Q \vee R$, которое не вынуждено в W_1 .

- 4 Написать программу, преобразующую вывод $\Gamma, \alpha \vdash \beta$ в вывод $\Gamma \vdash \alpha \rightarrow \beta$ (либо уведомляющую, что исходный вывод некорректен) в исчислении предикатов. Задание аналогично заданию 2, со следующими отличиями:

- Грамматика исчисления предикатов расширена следующим образом:

$$\begin{aligned}
 \langle \text{выражение} \rangle &::= \langle \text{дизъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle ' \rightarrow ' \langle \text{выражение} \rangle \\
 \langle \text{дизъюнкция} \rangle &::= \langle \text{конъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle ' \mid ' \langle \text{конъюнкция} \rangle \\
 \langle \text{конъюнкция} \rangle &::= \langle \text{унарное} \rangle \mid \langle \text{конъюнкция} \rangle ' \& ' \langle \text{унарное} \rangle \\
 \langle \text{унарное} \rangle &::= \langle \text{предикат} \rangle \mid ' ! ' \langle \text{унарное} \rangle \mid ' (' \langle \text{выражение} \rangle ') ' \\
 &\quad \mid ' @ ' \mid ' ? ' \langle \text{переменная} \rangle \langle \text{унарное} \rangle \\
 \langle \text{переменная} \rangle &::= (' a ' \dots ' z ') \{ ' 0 ' \dots ' 9 ' \}^* \\
 \langle \text{предикат} \rangle &::= (' A ' \dots ' Z ') \{ ' 0 ' \dots ' 9 ' \}^* [' (' \langle \text{терм} \rangle \{ ' , ' \langle \text{терм} \rangle \}^* ') '] \\
 \langle \text{терм} \rangle &::= (' a ' \dots ' z ') \{ ' 0 ' \dots ' 9 ' \}^* ' (' \langle \text{терм} \rangle \{ ' , ' \langle \text{терм} \rangle \}^* ') ' \\
 &\quad \mid \langle \text{переменная} \rangle \mid ' (' \langle \text{терм} \rangle ') '
 \end{aligned}$$

Символ '@' обозначает квантор всеобщности, символ '?' — квантор существования. Грамматика не различает переменные и константы; мы можем предполагать, что константы в формуле — это свободные переменные.

- Входной файл может содержать некорректный вывод — а именно, некоторые переходы в выводе могут быть корректными; входной файл по-прежнему синтаксически корректен. В этом случае программа должна выдать текст вида:

Вывод некорректен начиная с формулы номер <ℳ>[: <ошибка>]

где вместо $\langle \mathbb{N} \rangle$ подставлен номер первой некорректной формулы. Необязательное поле $\langle \text{ошибка} \rangle$ должно появляться, если формула не является аксиомой или допущением, не может быть выведена из предыдущих, но являлась бы (или могла бы быть выведена), если бы не нарушение ограничений на переменные. Возможные варианты ошибок:

терм $\langle X \rangle$ не свободен для подстановки в формулу $\langle Y \rangle$ вместо переменной $\langle a \rangle$.

переменная $\langle a \rangle$ входит свободно в формулу $\langle X \rangle$.

используется $\langle \text{правило} | \text{схема аксиом} \rangle$ с квантором по переменной $\langle a \rangle$, входящей свободно в допущение $\langle X \rangle$.

- 5 Написать программу, проверяющую доказательство в формальной арифметике на корректность.

Грамматика формул формальной арифметики:

$$\begin{aligned}
 \langle \text{выражение} \rangle &::= \langle \text{дизъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle ' \rightarrow ' \langle \text{выражение} \rangle \\
 \langle \text{дизъюнкция} \rangle &::= \langle \text{конъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle ' \vee ' \langle \text{конъюнкция} \rangle \\
 \langle \text{конъюнкция} \rangle &::= \langle \text{унарное} \rangle \mid \langle \text{конъюнкция} \rangle ' \& ' \langle \text{унарное} \rangle \\
 \langle \text{унарное} \rangle &::= \langle \text{предикат} \rangle \mid ' ! ' \langle \text{унарное} \rangle \mid ' (' \langle \text{выражение} \rangle ') ' \\
 &\quad \mid ' \emptyset ' \mid ' ? ' \langle \text{переменная} \rangle \langle \text{унарное} \rangle \\
 \langle \text{переменная} \rangle &::= (' a ' \dots ' z ') \{ ' 0 ' \dots ' 9 ' \}^* \\
 \langle \text{предикат} \rangle &::= (' A ' \dots ' Z ') \{ ' 0 ' \dots ' 9 ' \}^* [' (' \langle \text{терм} \rangle \{ ' , ' \langle \text{терм} \rangle \}^* ') '] \\
 &\quad \mid \langle \text{терм} \rangle ' = ' \langle \text{терм} \rangle \\
 \langle \text{терм} \rangle &::= \langle \text{слагаемое} \rangle \mid \langle \text{терм} \rangle ' + ' \langle \text{слагаемое} \rangle \\
 \langle \text{слагаемое} \rangle &::= \langle \text{умножаемое} \rangle \mid \langle \text{слагаемое} \rangle ' * ' \langle \text{умножаемое} \rangle \\
 \langle \text{умножаемое} \rangle &::= (' a ' \dots ' z ') \{ ' 0 ' \dots ' 9 ' \}^* ' (' \langle \text{терм} \rangle \{ ' , ' \langle \text{терм} \rangle \}^* ') ' \\
 &\quad \mid \langle \text{переменная} \rangle \mid ' (' \langle \text{терм} \rangle ') ' \\
 &\quad \mid ' 0 ' \mid \langle \text{умножаемое} \rangle ' , '
 \end{aligned}$$

От грамматики исчисления предикатов она отличается специальным синтаксисом для предикатов ($=$), функциональных символов и констант (\cdot , $+$, $'$, 0), использующихся в формальной арифметике.

Выходной файл — либо фраза

Доказательство корректно.

либо указание об ошибке в формате указания об ошибках для задачи 4.

- 6 Выразить через примитивы рекурсивных функций:

- сложение
- ограниченное вычитание ($' \dot{-} '$); $a \dot{-} b = \max(a - b, 0)$,
- целочисленное деление (div)
- остаток от деления (mod)
- проверку на простоту ($isPrime$); $isPrime(5) = 1, isPrime(9) = 0$
- вычисление n -го простого числа ($nthPrime$); $nthPrime(0) = 2$
- частичное логарифмирование ($plog(p, x)$) — максимальное целое число $r \geq 0$, такое, что x делится на p^r — например, $plog(3, 12) = 2, plog(5, 7) = 0$.

Решение задачи — это программа, содержащая код рекурсивных функций в синтаксисе, подобном синтаксису, использованному на лекциях (угловые скобки не обязательны, можно и круглые). Также, в данной программе должны быть приведены примеры их использования, вычисляющие результаты использования функций при некоторых значениях.

Рекурсивные функции можно реализовывать и не выходя за рамки какого-то языка программирования общего назначения (рекомендуемый вариант; скажем, это можно делать с помощью шаблонов C++, или с помощью комбинаторов в функциональных языках), либо с помощью отдельно написанного интерпретатора.

- 6.5 Реализовать в условиях задачи 6 рекурсивную функцию, вычисляющую функцию Аккермана. Решение данной задачи должно выдавать корректный ответ для первых чисел Аккермана ($A(1, 1)$, а лучше $A(2, 2)$) за разумное время (несколько минут) — пусть даже для этого потребуются подменить примитивно-рекурсивную реализацию арифметических функций на «нативную».

Подсказка: можно использовать подход, который применялся при поиске формулы, представляющей примитивную рекурсию. С β -функцией решение, впрочем, будет по всей видимости работать медленно, ее имеет смысл заменить на что-то более приспособленное.